AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions in the application.

Listing of Claims:

CLAIM 1 (Previously Presented): A software method for constructing a reconfigurable supercomputing virtual machine, and both local and distributed networks of such virtual machines, each having its own instruction set, and operating on an underlying physical hardware processor or processors, comprising operations that enable:

a) the virtual machine architecture to operate directly on a class of problems having a solution describable in terms of nodes;

b) whereby a node comprises an index word and a data word;

c) whereby each node represents one or more of the following data structures:

numeric tags, character tags, boolean flags, numeric values, character values, objects IDs, database-record IDs, simple arrays, variable-density multidimensional arrays, symbolic functions, mathematical functions, connection pointers to other nodes, function pointers, lookup-table list pointers, linked-lists, or even pointers to other solution spaces or data representations;

d) whereby nodes are interconnected in any of the following topologies: independent point-clouds, ordered sets of points, acyclic graphs, cyclic graphs, balanced trees, recombining graphs, meshes, lattices, and various hybrids or combinations of such representations;

e) whereby the virtual machine architecture includes units to configure nodes and virtual processor architecture, including register structures, node data structures, arithmetic modes, and memory schemes;

f) and virtual units to create (that is, instantiate) nodes, compute results for (that is, populate) nodes, move among (that is, navigate) nodes, and alter (that is, evolve) nodes and their interconnections;

g) a virtual unit to provide highly-optimized function evaluation and fixed-point integer arithmetic, with application-selectable precision;

h) a virtual unit to manage the distribution of data and processes to networked machines.

CLAIM 2 (Previously Presented): A software method that implements a multi-tasking operating system for the virtual machine of Claim 1, that allows for multiprocessing via multiple virtual machines implemented on a network of underlying hardware processors, in a local or distributed cluster, comprising operations that:

a) create a new virtual CPU for each task thread;

b) realize software engines for configuring, instantiating, populating, navigating, and modifying (evolving) nodes;

c) realize autonomous daemons for background processing of nodes;

d) realize a toolbox containing frequently-used engine programs;

e) realize an assembler for translating operating-system calls into virtual machine operation codes;

f) realize platform drivers for executing virtual-machine operations on the underlying physical platform processor;

g) realize a platform assembler for translating virtual machine operations into instruction-codes for platform operations.

CLAIM 3 (Previously Presented): A software method applied to the virtual machine of Claim 1, that allows for faster, and therefore lower cost, software application creation, and that produces computer programs that rapidly generate 'good enough' solutions to computationally complex and/or high-demand problems that are describable in terms of nodes, via a set of non-sequential processes, comprising:

a) matching the virtual machine architecture and solution manifold to the problem architecture;

b) adapting the solution manifold in response to changing demands in problem architecture or data;

c) adapting the virtual-machine architecture in response to changing demands in problem architecture or data;

d) using application-selectable arithmetic precision to rapidly compute 'accurate enough' calculations when evaluating nodes;

e) using software-emulation of supercomputing techniques, such as small instruction set, simple and efficient data representation and handling, inherent vector representation, limited data/calculation modes, interleaved memory, table lookup, induced pointers, and distributed & parallelized computation;

f) separating the populating and navigating of nodes, to allow for pre-computation of manifolds, so that navigation of possible solutions occurs in near real-time;

g) using autonomous, second-order dedicated processes that operate in background, as concurrent tasks, to collect garbage, prune trees, condense redundancies, process edit-queues, interpolate with finer granularity (mesh enhancement) around selected nodes in state-space, or to extrapolate and elaborate the data structures, during both population and navigation phases;

h) generating virtual CPUs for each operating-system task thread.

CLAIM 4 (New): A computer system for constructing a reconfigurable supercomputing virtual machine, and both local and distributed networks of such virtual machines, each having its own instruction set, the computer system comprising:

a first storage device for storing all the virtual machines created, and their associated operation specifications and data;

a processor connected to the first storage device, with the processor configured for operations that enable:

a) the virtual machine architecture to operate directly on a class of problems having a solution describable in terms of nodes;

b) whereby a node comprises an index word and a data word;

c) whereby each node represents one or more of the following data structures:

numeric tags, character tags, boolean flags, numeric values, character values, objects IDs, database-record IDs, simple arrays, variable-density multidimensional arrays, symbolic functions, mathematical functions, connection pointers to other nodes, function pointers, lookup-table list pointers, linked-lists, or even pointers to other solution spaces or data representations;

d) whereby nodes are interconnected in any of the following topologies: independent point-clouds, ordered sets of points, acyclic graphs, cyclic graphs, balanced trees, recombining graphs, meshes, lattices, and various hybrids or combinations of such representations;

e) whereby the virtual machine architecture includes units to configure nodes and virtual processor architecture, including register structures, node data structures, arithmetic modes, and memory schemes;

f) and virtual units to create (that is, instantiate) nodes, compute results for (that is, populate) nodes, move among (that is, navigate) nodes, and alter (that is, evolve) nodes and their interconnections;

g) a virtual unit to provide highly-optimized function evaluation and fixed-point integer arithmetic, with application-selectable precision;

h) a virtual unit to manage the distribution of data and processes to networked machines.

CLAIM 5 (New): A computer system that implements a multi-tasking operating system for the virtual machine(s) of Claim 4, that allows for multiprocessing via multiple virtual machines implemented on a network of underlying hardware processors, in a local or distributed cluster, the computer system comprising:

a first storage device for storing all the virtual machines created, their associated operation specifications and data, and the multi-tasking operating system;

a processor connected to the first storage device, with the processor configured for operations that enable the processor to:

a) create a new virtual CPU for each task thread;

b) realize software engines for configuring, instantiating, populating, navigating, and modifying (evolving) nodes;

c) realize autonomous daemons for background processing of nodes;

d) realize a toolbox containing frequently-used engine programs;

e) realize an assembler for translating operating-system calls into virtual machine operation codes;

f) realize platform drivers for executing virtual-machine operations on the underlying physical platform processor;

g) realize a platform assembler for translating virtual machine operations into instruction-codes for platform operations.

CLAIM 6 (New): A computer system to implement the virtual machine(s) of Claim 4, the computer system comprising:

a first storage device for storing all the virtual machines created, their associated operation specifications and data, the multi-tasking operating system, and all user-created applications and data;

a processor connected to the first storage device, with the processor configured for operations that enable the processor to allow for faster, and therefore lower cost, software application creation, and that produces computer programs that rapidly generate 'good enough' solutions to computationally complex and/or high-demand problems that are describable in terms of nodes, via a set of non-sequential processes, comprising:

a) matching the virtual machine architecture and solution manifold to the problem architecture;

b) adapting the solution manifold in response to changing demands in problem architecture or data;

c) adapting the virtual-machine architecture in response to changing demands in problem architecture or data;

d) using application-selectable arithmetic precision to rapidly compute 'accurate enough' calculations when evaluating nodes;

e) using software-emulation of supercomputing techniques, such as small instruction set, simple and efficient data representation and handling, inherent vector representation, limited data/calculation modes, interleaved memory, table lookup, induced pointers, and distributed & parallelized computation;

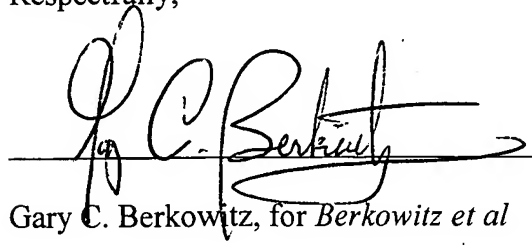
f) separating the populating and navigating of nodes, to allow for pre-computation of manifolds, so that navigation of possible solutions occurs in near real-time;

g) using autonomous, second-order dedicated processes that operate in background, as concurrent tasks, to collect garbage, prune trees, condense redundancies, process edit-queues, interpolate with finer granularity (mesh enhancement) around selected nodes in state-space, or to extrapolate and elaborate the data structures, during both population and navigation phases;

h) generating virtual CPUs for each operating-system task thread.

We submit the amended Claims for consideration, as properly allowing the claims to meet the enablement, written description, and distinct & definite requirements of 35 USC 112, the novelty requirements of 35 USC 102, and the non-obviousness requirements of 35 USC 103. We thus hereby request that the Claims be allowed.

Respectfully,



Gary C. Berkowitz, for *Berkowitz et al*